Under consideration for publication in Math. Struct. in Comp. Science

Characterizing co-NL by a Group Action

CLÉMENT AUBERT[1] and THOMAS SEILLER[2 †]

[1] *Université Paris 13, Sorbonne Paris Cité, LIPN, CNRS, UMR 7030,F-93430, Villetaneuse, France.*
[2] *LAMA — UMR 5127 CNRS - Université de Savoie*

Received September 18, 2012

In a recent paper (Girard 2011*b*), Girard uses the geometry of interaction in the hyperfinite factor (Girard 2011*a*) in an innovative way to characterize complexity classes. The purpose of this paper is two-fold: to give a detailed explanation of both the choices and the motivations of Girard's definitions, and – since Girard's paper skips over some non-trivial details and only sketches one half of the proof – to provide a complete proof that **co-NL** can be characterized by an action of the group of finite permutations. We introduce as a technical tool the non-deterministic pointer machine, a concrete model that computes the algorithms represented in this setting.

1. Introduction

Traditionally, the study of complexity relies on the definition of programs as Turing Machines, or some kind of abstract device. In recent years, a new approach to complexity stemmed from the so-called proof-as-program — or Curry-Howard — correspondence which allows to understand program execution as a cut-elimination procedure in logic. This correspondence naturally extends to quantitative approaches that made it possible to work on complexity with tools coming from logic. Due to its resource-awareness, Linear Logic (**LL**) is particularly suitable to treat computational questions, and many attempt have been made to characterize complexity classes using this formalism. To name a few, Elementary Linear Logic (**ELL**) (Danos & Joinet 2001), Soft Linear Logic (Lafont 2004) and Bounded Linear Logic (Dal Lago & Hofmann 2009) characterize complexity classes, but only deterministic, sequential and equal to **P** (polytime) or above. New directions have recently been explored to characterize other complexity classes: SBAL (Schöpp 2007) characterizes **L** (logarithmic space), Boolean Proof Nets (Terui 2004, Aubert 2011), was the first success toward a characterization of parallel classes.

All those attempts belong to the field of Implicit Computational Complexity (**ICC**) whose main advantage is that we don't have to refer to a particular model or an external measuring condition. We only have to consider language restrictions (for instance by limiting the primitive recursion) or to infer the complexity properties of a program, for instance with

† This work was partly supported by the ANR-10-BLAN-0213 Logoi.

techniques like quasi-interpretations. **LL** offers a particularly nice framework to study complexity questions since the decomposition of implication into a linear implication and a duplication modality allows some fine tuning of the rules that govern the latter. All the attempts previously quoted are *implicit* characterization of complexity classes, as those logical system rest on the limitation of the computational power of **LL**. Next to the restrictions of recursion and the rewriting system with quasi-interpretation, this approach exhibits several interesting results as we do not have to compute to know the limitations on space or time of our formalism.

The *geometry of interaction* program (Girard 1989) was introduced by Girard a few years after the introduction of **LL**. It aims at giving an interpretation of proofs — or programs — that accounts for the dynamics of cut-elimination, hence of computation. By using realizability techniques, the different constructions introduced throughout the years allow to reconstruct types from the way programs interact with each other. Due to the fact that they are centered around the notion of computation, these constructions are particularly adapted to study computational complexity (Baillot, Pedicini et al. 2001, Lago 2005).

The approach studied in this paper, which was proposed recently by Girard, differs from the previous works on complexity. Indeed, though it uses the tools of Girard's geometry of interaction in the hyperfinite factor (Girard 2011*a*), its relation to the latter is restricted to the representation of integers which is, in this particular setting, uniform: each integer is represented as an operator in the hyperfinite type II_1 factor $\mathfrak{R}$. By using an operator-theoretic construction — the crossed product — it is possible to internalize some isomorphisms acting on $\mathfrak{R}$. Seeing each operator ϕ in the subalgebra generated by these internalized isomorphisms as a sort of abstract machine, one can associate to it a subset of the natural numbers: those numbers whose representation composed with ϕ yields a nilpotent linear map, that is the natural numbers for which the machine ϕ halts. In this paper, we will present in details a first result obtained from this approach: considering the group of finite permutations of the natural numbers, we can obtain a characterization of the complexity class **co-NL**. To ease the presentation and proofs of the result, we will introduce *non-deterministic pointer machines*, which are a new characterization of **co-NL** in terms of abstract machines.

2. Binary Integers

In this paper, we will be working with binary integers. In this section, we will explain how it is possible to represent these integers by matrices. As it turns out, representation by matrices is not satisfactory, and it will be necessary to represent integers by operators acting on an infinite-dimensional (separable) Hilbert space.

In intuitionistic logic, binary lists are typed with $\forall X\ (X \to X) \to ((X \to X) \to (X \to X))$. In **ELL**, the type of binary lists is:

$$\forall X\ !(X \multimap X) \multimap (!(X \multimap X) \multimap !(X \multimap X))$$

To a binary integer corresponds a proof of the sequent $\vdash ?(X \otimes X^{\perp}), ?(X \otimes X^{\perp}), !(X \multimap X)$. Whereas the conclusion is always the same, the number of contraction and weakening rules and the occurrences of formulas which are contracted or weakened may differ from

one proof to another. We develop below three examples: the empty list <> and the lists < 0 > and < 110 >. In the sequent calculus proofs below the variables are labeled by some information appearing in parentheses because one need to distinguish between occurrences of variables to keep track of the principal formulas in a contraction. This distinction, that appears in geometry of interaction and needs an explicit labelling of the variables to be understood, is crucial since it may be the only difference between the proofs corresponding to two different binary lists. For instance, without this information, the proofs representing < 110 > and < 010 > would be exactly the same.

To each sequent calculus proof, we associate a graph which represents the axiom links in the sequent calculus proof. The vertices represent the different occurrences of the variables in the conclusion horizontally, and a number of *slices* vertically: the contraction is represented in geometry of interaction by a superimposition which is dealt with by introducing new copies of the occurrences. In previous works (Seiller 2011, Seiller 2012*a*), one of the authors showed how to obtain a combinatorial version of (a fragment) of Girard's geometry of interaction in the hyperfinite factor. Though the graphs shown here are more complex than the ones considered in these papers (in particular, the edges may go from one *slice* to an other), they correspond exactly to the representation of binary lists in Girard framework[†].

- The proof representing the empty list uses the weakening rule twice:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\vdash X(S)^{\perp}, X(E)}\,\text{ax}}{\vdash X(S) \multimap X(E)}\,⅋}{\vdash !(X(S) \multimap X(E))}\,!}{\vdash ?(X(0i) \otimes X(0o)^{\perp}), !(X(S) \multimap X(E))}\,?w}{\vdash ?(X(0i) \otimes X(0o)^{\perp}), ?(X(1i) \otimes X(1o)^{\perp}), !(X(S) \multimap X(E))}\,?w}{\vdash !(X(0i) \multimap X(0o)) \multimap (!(X(1i) \multimap X(1o)) \multimap !(X(S) \multimap X(E)))}}{\vdash \forall X\ !(X(0i) \multimap X(0o)) \multimap (!(X(1i) \multimap X(1o)) \multimap !(X(S) \multimap X(E)))}\,\forall$$

The corresponding graph is:

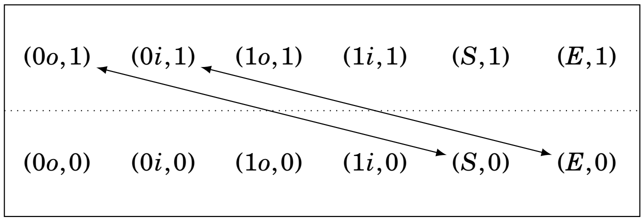
$$\boxed{\begin{tikzcd}[column sep=small] (0o,0) & (0i,0) & (1o,0) & (1i,0) & (S,0) \arrow[r, bend left, leftrightarrow] & (E,0)\end{tikzcd}}$$

- The proof representing the list < 0 > (resp. < 1 >) uses a weakening to introduce $X(1i) \multimap X(1o)$ (resp. $X(0i) \multimap X(0o)$):

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\vdash X(S)^{\perp}, X(0i)}\,\text{ax} \qquad \dfrac{}{\vdash X(0o)^{\perp}, X(E)}\,\text{ax}}{\vdash X(0i) \otimes X(0o)^{\perp}, X(S)^{\perp}, X(E)}\,\otimes}{\vdash X(0i) \otimes X(0o)^{\perp}, X(S) \multimap X(E)}\,⅋}{\vdash ?(X(0i) \otimes X(0o)^{\perp}), !(X(S) \multimap X(E))}\,!}{\vdash ?(X(0i) \otimes X(0o)^{\perp}), ?(X(1i) \otimes X(1o)^{\perp}), !(X(S) \multimap X(E))}\,?w}{\vdash !(X(0i) \multimap X(0o)) \multimap (!(X(1i) \multimap X(1o)) \multimap !(X(S) \multimap X(E)))}}{\vdash \forall X\ !(X(0i) \multimap X(0o)) \multimap (!(X(1i) \multimap X(1o)) \multimap !(X(S) \multimap X(E)))}\,\forall$$

The corresponding graph is:

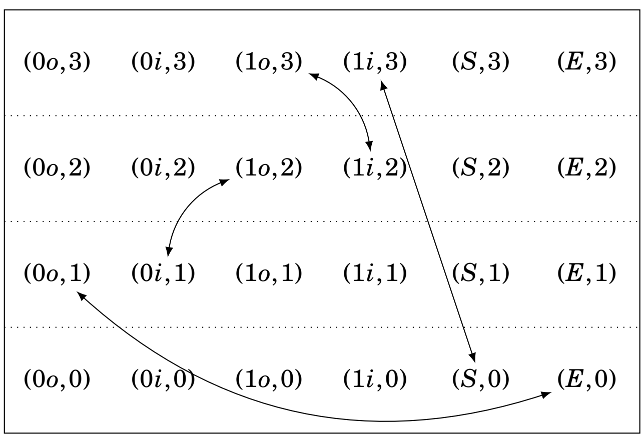
[†] For more details, one may consult Seiller's PhD Thesis (Seiller 2012*b*).



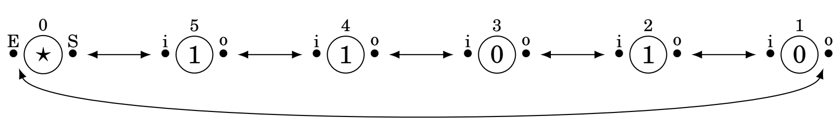
- The proof representing the list $< 110 >$ contracts the occurrences $X(Ai) \otimes X(Ao)^{\perp}$ and $X(1i) \otimes X(1o)^{\perp}$, in bold below:

$$
\begin{prooftree}
\AxiomC{}
\RightLabel{ax}
\UnaryInfC{$\vdash X(0i), X(E)^{\perp}$}
\AxiomC{}
\RightLabel{ax}
\UnaryInfC{$\vdash X(1i), X(0o)^{\perp}$}
\RightLabel{$\otimes$}
\BinaryInfC{$\vdash X(0i) \otimes X(0o)^{\perp}, X(1i), X(E)^{\perp}$}
\AxiomC{}
\RightLabel{ax}
\UnaryInfC{$\vdash X(Ai), X(1o)^{\perp}$}
\RightLabel{$\otimes$}
\BinaryInfC{$\vdash X(0i) \otimes X(0o)^{\perp}, X(1i) \otimes X(1o)^{\perp}, X(Ai), X(E)^{\perp}$}
\AxiomC{}
\RightLabel{ax}
\UnaryInfC{$\vdash X(S), X(Ao)^{\perp}$}
\RightLabel{$\otimes$}
\BinaryInfC{$\vdash X(0i) \otimes X(0o)^{\perp}, X(1i) \otimes X(1o)^{\perp}, X(Ai) \otimes X(Ao)^{\perp}, X(S), X(E)^{\perp}$}
\RightLabel{$\parr$}
\UnaryInfC{$\vdash X(0i) \otimes X(0o)^{\perp}, X(1i) \otimes X(1o)^{\perp}, X(Ai) \otimes X(Ao)^{\perp}, X(S) \multimap X(E)$}
\RightLabel{!}
\UnaryInfC{$\vdash ?(X(0i) \otimes X(0o)^{\perp}), ?(\mathbf{X(1i)} \otimes \mathbf{X(1o)})^{\perp}, ?(\mathbf{X(Ai)} \otimes \mathbf{X(Ao)})^{\perp}, !(X(S) \multimap X(E))$}
\RightLabel{$?_c$}
\UnaryInfC{$\vdash ?(X(0i) \otimes X(0o)^{\perp}), ?(\mathbf{X(1i)} \otimes \mathbf{X(1o)})^{\perp}, !(X(S) \multimap X(E))$}
\doubleLine
\UnaryInfC{$\vdash \forall X\ !(X(0i) \multimap X(0o)) \multimap (!(X(1i) \multimap X(1o)) \multimap !(X(S) \multimap X(E)))$}
\end{prooftree}
$$

  The corresponding graph is:

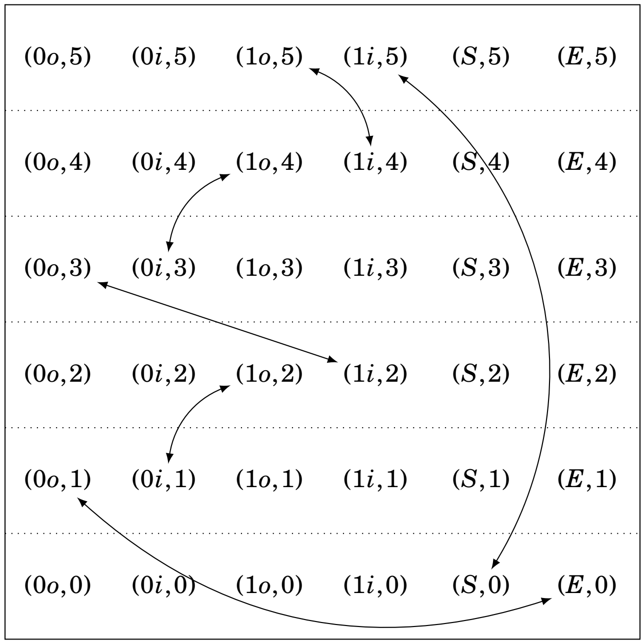


The edges of the graphs describe the scanning of the list: the edge starting from $X(S)$ goes to $X(0i)$ (resp. $X(1i)$) when the list start with a 0 (resp. a 1). The next edges goes from $X(0o)$ (resp. $X(1o)$) to $X(0i)$ if the next element of the list is a 0, to $X(1i)$ if the next element is a 1. It explains our notation: $X(0i)$ (resp. $X(1i)$) is the input of a 0 (resp. of a 1) and $X(0o)$ (resp. $X(1o)$) is the output of a 0 (resp. of a 1). We can construct the corresponding graph directly from the list, as for the next figure that represents the graph of $< 11010 >$:



Each element of the list lives in a different *slice* — the integer shown above each element of the list. Moreover, each element is connected by its output node to its successor's input

node (the successor of the last element is $\star$), and by its input node to its predecessor's output node. This gives the following graph, which is the representation of $< 11010 >$:



Definition 1 (Matricial representation of a list)**.** Given a list $a =< a_1,\dots,a_n >$ of size $n \neq 0$ and its corresponding graph G_a, a is represented by N_n a 6×6 bloc matrices of the following form:

$$
\begin{array}{c}
\overbrace{\hphantom{0 \quad l_{00}}}^{0} \quad \overbrace{\hphantom{0 \quad l_{10}}}^{1} \quad \overbrace{\hphantom{s_0 \quad 0}}^{*} \\
\left(\begin{array}{cccccc}
0 & l_{00} & 0 & l_{10} & s_0 & 0 \\
l_{00}^{*} & 0 & l_{01}^{*} & 0 & 0 & e_0^{*} \\
0 & l_{01} & 0 & l_{11} & s_1 & 0 \\
l_{10}^{*} & 0 & l_{11}^{*} & 0 & 0 & e_1^{*} \\
s_0^{*} & 0 & s_1^{*} & 0 & 0 & 0 \\
0 & e_0 & 0 & e_1 & 0 & 0
\end{array}\right)
\begin{array}{l}
\left.\vphantom{\begin{array}{c}0\\0\end{array}}\right\}0 \\
\left.\vphantom{\begin{array}{c}0\\0\end{array}}\right\}1 \\
\left.\vphantom{\begin{array}{c}0\\0\end{array}}\right\}*
\end{array}
\end{array}
$$

where coefficients are $(n+1)\times(n+1)$ matrices (the $(\cdot)^*$ denotes the conjugate-transpose) defined by:

- for $j,k \in \{0,1\}$, $(l_{jk})_{a,b} = 1$ if there is an edge in G_a from (jo,a) to (ki,b), and $(l_{jk})_{a,b} = 0$ otherwise;
- for $j \in \{0,1\}$, $(s_j)_{0,n} = 1$ (resp. $(e_j)_{0n} = 1$) if there is an edge in G_a between $X(S)$ and $X(jn)$ (resp. from $X(E)$ to $X(jo)$), and $(s_j)_{a,b} = 0$ (resp. $(e_j)_{a,b} = 0$) otherwise.

The embraces are here to help the reader to read this matrix: we can easily recover where are the information regarding the edges going for instance from a 1 to a 0.

This representation of binary integers is however non-uniform: the size of the matrix depends on the size of the list of the represented integer. This is where the use of von

Neumann algebra takes its importance: any matrix algebra can be embedded in the type II_1 hyperfinite factor $\mathfrak{R}$. In order to get a uniform representation of integers, we therefore only need to embed the matricial representation in $\mathfrak{R}$. Before explaining this step, we review in the next section some basics of the theory of von Neumann algebras. The aim of this section is not to introduce the reader to the theory which is much too rich to be condensed here, but to give some ideas and intuitions on it. In the end of the next section, we introduce the crossed product of an algebra with a group, an operation which is the main tool that will be used in this paper.

3. von Neumann Algebras and Crossed Products

3.1. *Hilbert Spaces and Operators*

We consider the reader familiar with the notions of Hilbert spaces and operators (continuous — or equivalently bounded — linear maps between Hilbert spaces). We refer to the classic textbooks of Conway (Conway 1990) for the bases of the theory, and Murphy's (Murphy 1990) for an excellent introduction to the theory of operator algebras. We will not dwell on the definitions and properties of von Neumann algebras, factors, and hyperfiniteness. We believe all these notions, though used in this paper and in Girard's, are not at the core of the characterization, and will not play an important rôle in the following construction. We therefore refer to the series of Takesaki (Takesaki 2001, Takesaki 2003*a*, Takesaki 2003*b*). A quick overview of the needed material can also be found in the appendix of the paper of one the authors (Seiller 2011).

We recall that an operator T is a linear map from $\mathbb{H}$ — a Hilbert space — to $\mathbb{H}$ that is continuous. A standard result tells us that this is equivalent to T being bounded, i.e. that there exists a constant C such that for all $\xi \in \mathbb{H}$, $\|T\xi\| \leqslant C\|\xi\|$. The smallest such constant defines a norm on $\mathscr{L}(\mathbb{H})$ — the set of operators on $\mathbb{H}$ —which we will denote by $\|T\|$.

Being given an operator T in $\mathscr{L}(\mathbb{H})$, we can show the existence of its *adjoint* — denoted by T^* —, the operator that satisfies $<T\xi,\eta> = <\xi,T^*\eta>$ for all $\xi,\eta \in \mathbb{H}$. It is easily shown that $T^{**} = T$, i.e. that $(\cdot)^*$ is an involution, and that it satisfies the following conditions:

1 For all $\lambda \in \mathbb{C}$ and $T \in \mathscr{L}(\mathbb{H})$, $(\lambda T)^* = \bar{\lambda}T^*$;
2 For all $S,T \in \mathscr{L}(\mathbb{H})$, $(S+T)^* = S^* + T^*$;
3 For all $S,T \in \mathscr{L}(\mathbb{H})$, $(ST)^* = T^*S^*$.

In a Hilbert space $\mathbb{H}$ there are two natural topologies, the topology induced by the norm on $\mathbb{H}$, and a weaker topology defined by the inner product.

1 The strong topology: we say a sequence $\{\xi_i\}_{i\in\mathbf{N}}$ converges strongly to 0 when $\|\xi_i\| \to 0$.
2 The weak topology: a sequence $\{\xi_i\}_{i\in\mathbf{N}}$ converges weakly to 0 when $<\xi_i,\eta> \to 0$ for all $\eta \in \mathscr{L}(\mathbb{H})$. Weak convergence is thus a point-wise or direction-wise convergence.

On $\mathscr{L}(\mathbb{H})$, numerous topologies can be defined, each of which having its own advantages and drawbacks. The five most important topologies are the norm topology, the strong operator topology, the weak operator topology, the ultra-strong (or σ-strong) topology and the ultra-weak (or σ-weak) topology. We can easily characterize the first three topologies in terms of converging sequences as follows:

1 The norm topology: $\{T_i\}_{i\in\mathbf{N}}$ converges (for the norm) to 0 when $\|T_i\| \to 0$;
2 The strong operator topology, which is induced by the strong topology on $\mathbb{H}$: $\{T_i\}_{i\in\mathbf{N}}$ converges strongly to 0 when, for any $\xi \in \mathbb{H}$, $T_i\xi$ converges strongly to 0 ;
3 The weak operator topology, which is induced by the weak topology on $\mathbb{H}$: $\{T_i\}_{i\in\mathbf{N}}$ converges weakly to 0 when, for any $\xi \in \mathbb{H}$, $T_i\xi$ converges weakly to 0.

We can show that $\mathscr{L}(\mathbb{H})$ is the dual of a space denoted by $\mathscr{L}(\mathbb{H})_*$ containing the *trace-class operators*. For further details, the reader may refer to (Murphy 1990) or (Takesaki 2001). We remind here of this result only to define the σ-weak topology: if A is a topological space and A^* is its dual, the *weak* topology* on A is defined as the point-wise topology.

3.2. *von Neumann Algebras in a Nutshell*

Let $\mathbb{H}$ be a Hilbert space, and $\mathscr{L}(\mathbb{H})$ be the set of bounded — continuous — linear maps from $\mathbb{H}$ to itself. It is standard knowledge that $\mathscr{L}(\mathbb{H})$ is an associative algebra when endowed with composition and pointwise scalar multiplication and addition. It is moreover a complete normed vector space for the operator norm, defined as $\|u\| = \sup\{x \in \mathbb{H} \mid \|u(x)\|/\|x\|\}$. It is therefore what is called a Banach algebra. On the other hand, it is known that every element of $\mathscr{L}(\mathbb{H})$ has an *adjoint* operator u^*. This operation $(\cdot)^*$ is an involution satisfying: $(t+u)^* = t^* + u^*$, $(tu)^* = u^*t^*$, $(\lambda u)^* = \bar{\lambda}u^*$, $\|u^*\| = \|u\|$, and $\|u^*u\| = \|u\|^2$. A Banach algebra endowed with such an involution is called a C^*-algebra. As it turns out (this is the famous Gelfand-Naimark-Segal (GNS) construction), any C^*-algebra can be represented as a norm-closed $*$-subalgebra of $\mathscr{L}(\mathbb{H})$ for a Hilbert space $\mathbb{H}$.

A von Neumann algebra $\mathfrak{M}$ is a C^*-subaglebra of $\mathscr{L}(\mathbb{H})$, where $\mathbb{H}$ is a Hilbert space, which is closed for a weaker topology than the norm topology: the strong-operator topology, which is pointwise convergence on $\mathbb{H}$ considered with its norm topology. The first important result of the theory, obtained by von Neumann, is that this requirement is equivalent to the requirement that $\mathfrak{M}$ is closed for the even weaker *weak operator topology* which is pointwise convergence on $\mathbb{H}$ considered with its weak — "direction-wise" — topology. It is also equivalent to a completely algebraic condition which is the fact that $\mathfrak{M}$ is equal to its bi-commutant: let us denote $\mathfrak{M}'$ — the commutant of $\mathfrak{M}$ — the set of elements of $\mathscr{L}(\mathbb{H})$ which commute with every element of $\mathfrak{M}$, then $\mathfrak{M}''$ denotes the bi-commutant of $\mathfrak{M}$, that is the commutant of the commutant of $\mathfrak{M}$.

The study of von Neumann algebras was quickly reduced to the study of factors, that is von Neumann algebras $\mathfrak{M}$ whose center — the algebra of elements commuting with every element of $\mathfrak{M}$— is trivial: i.e. von Neumann algebras $\mathfrak{M}$ such that $\mathfrak{M}\cap\mathfrak{M}' = \mathbf{C}1_{\mathfrak{M}}$. Indeed, any von Neumann algebra can be decomposed along its center as a direct integral (a continuous direct sum) of factors. Factors $\mathfrak{N}$ can then be easily classified by their sets of projections (operators p such that $p = p^* = p^2$):

— **Type I:** $\mathfrak{N}$ contains non-zero finite minimal projections. If the identity of $\mathfrak{N}$ is the sum of a finite number — say n — of minimal projections, $\mathfrak{N}$ is of type I_n, and if it is not the case $\mathfrak{N}$ is of type I_∞.
— **Type II:** $\mathfrak{N}$ contains finite projections but has no minimal projections. Then if the identity of $\mathfrak{N}$ is a finite projection, $\mathfrak{N}$ is of type II_1, and it is of type II_∞ otherwise.
— **Type III:** all the non-zero projections of $\mathfrak{N}$ are infinite.

Among von Neumann algebras, the approximately finite dimensional ones are of particular interest, and are usually called *hyperfinite*. These are algebras in which every operator can be approximated (in the sense of the σ-weak topology[‡]) by a sequence of finite-dimensional operators (elements of type I_n factors, for $n \in \mathbf{N}$). In particular, the type II_1 hyperfinite factor is unique up to isomorphism (in fact, for most types hyperfinite factors are unique).

The definition we gave of von Neumann algebras is a concrete definition, i.e. as an algebra of operators acting on a Hilbert space. It turns out that von Neumann algebras can be defined abstractly as C^*-algebras that are the dual space of a Banach space. In the next subsection, and more generally in this paper, the term "von Neumann algebra" will have the meaning of "abstract von Neumann algebra".

3.3. *von Neumann Algebras and Groups*

Definition 2 (Representations)**.** Let $\mathfrak{M}$ be a von Neumann algebra. A couple $(\mathbb{H}, \rho)$ where $\mathbb{H}$ is a Hilbert space and ρ is a $*$-homomorphism from $\mathfrak{M}$ to $\mathscr{L}(\mathbb{H})$ is a representation of $\mathfrak{M}$. If ρ is injective, we say the representation is *faithful*.

Among the numerous representations of a von Neumann algebra, one can prove the existence (Haagerup 1975) of the so-called *standard representation*, a representation satisfying several important properties.

The operation that will be of interest to us will be that of taking the *crossed product* of an algebra and a group. This operation is closely related to that of semi-direct product of groups and is a way of internalizing automorphisms. Given an algebra $\mathfrak{A}$ and a group G of automorphisms of $\mathfrak{A}$, we construct the algebra $\mathfrak{A} \rtimes G$ generated by the elements of $\mathfrak{A}$ and the elements of G.

Definition 3. An action of a topological group G on a von Neumann algebra $\mathscr{M}$ is a continuous homomorphism of G into $\mathrm{Aut}(\mathscr{M})$.

Definition 4 (Crossed product (representations))**.** Let $(\mathbb{H}, \rho)$ be a representation of a von Neumann algebra $\mathfrak{M}$, G a locally compact group, and α an action of G on $\mathscr{M}$. Let $\mathbb{K} = L^2(G, \mathbb{H})$ be the Hilbert space of square-summable $\mathbb{H}$-valued functions on G. We define representations π_α of $\mathscr{M}$ and λ of G on $\mathbb{K}$ as follows

$$\begin{array}{rcl}(\pi_\alpha(x).\xi)(g) & = & (\rho(\alpha(g)^{-1}(x))\xi(g) \\ (\lambda(g).\xi)(h) & = & \xi(g^{-1}h)\end{array}$$

Then the von Neumann algebra on $\mathbb{K}$ generated by $\pi_\alpha(\mathscr{M})$ and $\lambda(G)$ is called the crossed product of $(\mathbb{H}, \rho)$ by α.

An important fact is that the result of the crossed product does not depend on the chosen representation of $\mathfrak{M}$. The following theorem, which states this fact, will be of use in a technical lemma at the end of this paper.

[‡] In a nutshell, the algebra $\mathscr{L}(\mathbb{H})$ is the dual of the algebra of *trace-class operators*. As a dual, it thus inherits the traditional weak* topology, which is called in the context of von Neumann algebras the σ-weak topology.

Theorem 5 (Takesaki (Takesaki 2003*a*), Theorem 1.7 page 241)**.** *Let* $(\mathbb{H},\rho)$ *and* $(\mathbb{K},\rho')$ *be two faithful representations of a von Neumann algebra* $\mathfrak{M}$*, and let* G *be a locally compact group together with an action* α *on* $\mathfrak{M}$*. Then there exists an isomorphism between the crossed product of* $(\mathbb{H},\rho)$ *by* α *and the crossed product* $(\mathbb{K},\rho')$ *by* α*.*

As a consequence, one can define *the* crossed product of a von Neumann algebra and a group acting on it by choosing a particular representation. Of course, the natural choice is to consider the *standard representation*.

Definition 6 (Crossed product)**.** Let $\mathfrak{M}$ be a von Neumann algebra, G a group and α an action of G on $\mathfrak{M}$. The algebra $\mathfrak{M} \rtimes_\alpha G$ is defined as the crossed product of the standard representation of $\mathfrak{M}$ by α.

A particular case of crossed product is the crossed product of $\mathbf{C}$ by a (trivial) action of a group G. The resulting algebra is usually called the group von Neumann algebra $\mathfrak{N}(G)$ of G. As it turns out, the operation of internalizing automorphisms of algebras (the crossed product) and the operation of internalizing automorphisms of groups (the semi-direct product) correspond: the algebra $\mathfrak{N}(G \rtimes_\alpha H)$ is isomorphic to $\mathfrak{N}(G) \rtimes_{\tilde{\alpha}} H$ where $\tilde{\alpha}$ is the action of H on $\mathfrak{N}(G)$ induced by the action of H on G.

4. Integers in the Hyperfinite Factor

4.1. *Binary Representation*

We will embed the $(n+1)\times(n+1)$ matrices of Definition 1 in the hyperfinite factor $\mathfrak{R}$ in order to have a uniform representation of the lists: a list will be represented by an operator in $\mathfrak{M}_6(\mathfrak{R})$ fulfilling some properties. To express them we define, given a sequence $< a_1,\dots,a_k >$ representing an integer n[§] and for $j,l \in \{0,1\}$, the sets:

$$\begin{aligned} I^n_{jl} &= \{1 \leqslant i \leqslant k \mid a_i = j, a_{i+1} = l\} \\ I^n_{Sj} &= \{i = 1 \mid a_i = j\} \\ I^n_{jE} &= \{i = k \mid a_i = j\} \end{aligned}$$

Roughly speaking, I^n_{Sj} (resp. I^n_{jE}) tells us about the first (resp. last) bit of our list and I^n_{jl} is the set of sequences of a j followed by a l.

Definition 7 (Binary representation of integers)**.** An operator $N_n \in \mathfrak{M}_6(\mathfrak{R})$ is a *binary representation* of an integer n if there exists projections $\pi_0,\pi_1,\dots,\pi_k$ in $\mathfrak{R}$ that satisfy $\sum_{i=0}^k \pi_i = 1$ such that:

$$u = \begin{pmatrix} 0 & l_{00} & 0 & l_{10} & l_{S0} & 0 \\ l^*_{00} & 0 & l^*_{01} & 0 & 0 & l^*_{0E} \\ 0 & l_{01} & 0 & l_{11} & l_{S1} & 0 \\ l^*_{10} & 0 & l^*_{11} & 0 & 0 & l^*_{1E} \\ l^*_{S0} & 0 & l^*_{S1} & 0 & 0 & 0 \\ 0 & l_{0E} & 0 & l_{1E} & 0 & 0 \end{pmatrix}$$

[§] We will always assume in the following that the length of the binary list representing the integer n under study is k.

where the coefficients are partial isometries fulfilling the equations (where $\pi_{k+1} = \pi_0$):

$$\begin{aligned} l_\star &= \sum_{i\in I^n_\star} \pi_{i+1} l_\star \pi_i \quad (\star \in \{00, 01, 10, 11, S0, S1, 0E, 1E\}) \\ \pi_0 &= (l_{0E} + l_{1E})(l_{00} + l_{01} + l_{10} + l_{11})^{k-1}(l_{S0} + l_{S1}) \end{aligned}$$

Proposition 8 (Binary and matricial representations)**.** *Given $N_n \in \mathfrak{M}_6(\mathfrak{R})$ a binary representation of the integer n, there exists an embedding $\phi : \mathfrak{M}_{k+1}(\mathbf{C}) \to \mathfrak{R}$ such that*[¶] *$Id \otimes \phi(M_n) = N_n$, and M_n is the matricial representation of the corresponding list.*

Proof. Let $N_n \in \mathfrak{R}$ a binary representation of $n \in \mathbf{N}$, and $\pi_0, \dots, \pi_k$ the associated projections. Notice that the projections π_i are pairwise equivalent.

We now define an embedding $\phi : \mathfrak{M}_{n+1}(\mathbf{C}) \to \mathfrak{R}$:

$$\phi : (a_{i,j})_{0\leqslant i,j\leqslant n} \mapsto \sum_{i=0}^{n}\sum_{j=0}^{n} a_{i,j} u_{i,j}$$

with:

$$u_{i,j} = \begin{cases} (l_{00} + l_{01} + l_{10} + l_{11})^{j-1}(l_{S0} + l_{S1}) & \text{if } i = 0 \\ (l_{00} + l_{01} + l_{10} + l_{11})^{j-1} & \text{if } i < j \text{ and } i \neq 0 \\ ((l_{00} + l_{01} + l_{10} + l_{11})^{i-1}(l_{S0} + l_{S1}))^* & \text{if } j = 0 \\ ((l_{00} + l_{01} + l_{10} + l_{11})^{i-1})^* & \text{if } i > j \text{ and } j \neq 0 \\ \pi_k & \text{if } i = j = k \end{cases}$$

We can easily check that the image by $\mathrm{Id} \otimes \phi$ of the matrix M_n representing n is equal to N_n. □

Proposition 9 (Equivalence of binary representations)**.** *Given N_n and N'_n two binary representations of $n \in \mathbf{N}$, there exists a unitary $u \in \mathfrak{R}$ such that $(Id \otimes u)N_n(Id \otimes u)^* = N'_n$.*

Proof. Let $\pi_0, \dots, \pi_n$ (resp. $\nu_0, \dots, \nu_n$) be the projections and $l_\star$ (resp. $l'_\star$) the partial isometries associated to N_n (resp. N'_n). It is straightforward that π_0 and ν_0 are equivalent according to Murray and von Neumann definition, so there exists a partial isometry v such that $vv^* = \nu_0$ and $v^*v = \pi_0$. For all $0 \leqslant i \leqslant n$ we define the partial isometries:

$$v_i = ((l'_{00} + l'_{01} + l'_{10} + l'_{11})^{i-1}(l'_{S0} + l'_{S1}))v((l_{00} + l_{01} + l_{10} + l_{11})^{i-1}(l_{S0} + l_{S1}))^*$$

We can easily check that:

$$\begin{aligned} v_i v_i^* &= \nu_i \\ v_i^* v_i &= \pi_i \end{aligned}$$

It follows that the sum $u = \sum_{i=0}^{n} v_i$ is a unitary and $(\mathrm{Id} \otimes u)N_n(\mathrm{Id} \otimes u)^* = N'_n$. □

[¶] We denote by Id the identity matrix of $\mathfrak{M}_6(\mathbf{C})$. We will allow ourselves the same abuse of notations in the following statements and proofs in order to simplify the formulas.

4.2. *Normative Pairs*

The notion of *normative pair*, a pair of two subalgebras $(\mathfrak{N},\mathfrak{O})$, was defined by Girard (Girard 2011*b*) in order to describe the kind of situations where the interaction of an operator in $\mathfrak{O}$ with two representations of the same integer in $\mathfrak{N}$ is sure to yield the same result. The interaction used by Girard was based on Fuglede-Kadison determinant[‖]. As a matter of fact, Girard defines his interaction with the determinant but actually uses nilpotency in his proofs. In order to give more flexibility to the definitions, we chose to work with an interaction based on nilpotency, which represents the fact the computation ends. This change in definition does not modify the fact that one can characterize **co-NL**, but allows one to consider a broader class of groups[††], and a broader class of languages[‡‡].

Definition 10 (Normative Pairs)**.** Let $\mathfrak{N}$ and $\mathfrak{O}$ be two subalgebras of a von Neumann algabra $\mathfrak{M}$. The pair $(\mathfrak{N},\mathfrak{O})$ is a *normative pair (in $\mathfrak{M}$)* if:

— $\mathfrak{N}$ is isomorphic to $\mathfrak{R}$;
— For all $\Phi \in \mathfrak{M}_6(\mathfrak{O})$ and $N_n, N'_n \in \mathfrak{M}_6(\mathfrak{N})$ two binary representations of n,

$$\Phi N_n \text{ is nilpotent} \Leftrightarrow \Phi N'_n \text{ is nilpotent}$$

Proposition 11. *Let S be a set and for all $s \in S$, $\mathfrak{N}_s = \mathfrak{R}$. For all group G and all action α of G on S, the algebra $\mathfrak{M} = (\bigotimes_{s\in S} \mathfrak{N}_s) \rtimes_{\hat{\alpha}} G$ contains a subalgebra generated by G that we will denote $\mathfrak{G}$. Then for all $s \in S$, the pair $(\mathfrak{N}_s, \mathfrak{G})$ is a normative pair (in $\mathfrak{M}$).*

Proof. From the hypotheses, $\mathfrak{N}_s$ is isomorphic to $\mathfrak{R}$. Regarding the second condition, we will only show one implication, the other being obtained by symmetry. By Lemma 9, there exists a unitary u such that $(\mathrm{Id}\otimes u)N_n(\mathrm{Id}\otimes u)^* = N'_n$. We define $v = \bigotimes_{s\in S} u$ and π_v the unitary in $\mathfrak{M}$ induced by v. Then π_v commutes with the elements of $\mathfrak{G}$, so if there exists $k \in \mathbf{N}$ such that $(\phi N_n)^k = 0$, then $(\phi N'_n)^k = (\phi u N_n u^*)^k = (u\phi N_n u^*)^k = u(\phi N_n)^k u^* = 0$. □

Definition 12 (Observations)**.** Let $(\mathfrak{N}_0, \mathfrak{S})$ be a normative pair. An *observation* is an operator in $\mathfrak{M}_6(\mathfrak{S}) \otimes \mathfrak{Q}$, where $\mathfrak{Q}$ is a matrix algebra, i.e. $\mathfrak{Q} = \mathfrak{M}_k(\mathbf{C})$ for an integer k, called the *algebra of states*.

Definition 13. Let $(\mathfrak{N}_0, \mathfrak{S})$ be a normative pair, and ϕ an observation. We define the set of natural numbers:

$$[\phi] = \{n \in \mathbf{N} \mid \phi N_n \text{ is nilpotent, } N_n \text{ a binary representation of } n\}$$

Definition 14. Let $(\mathfrak{N}_0, \mathfrak{S})$ be a normative pair and $X \subset \cup_{i=1}^{\infty} \mathfrak{M}_6(\mathfrak{S}) \otimes \mathfrak{M}_i(\mathbf{C})$ be a set of observations. We define the *language decided by X* as the set:

$$\{X\} = \{[\phi] \mid \phi \in X\}$$

[‖] A generalization of the usual determinant of matrices that can be defined in a type II_1 factor.

[††] The use of the determinant forces Girard to consider only amenable groups, so that the result of the crossed product in Proposition 11 yields the type II_1 hyperfinite factor.

[‡‡] In this paper and in Girard's, we consider languages obtained from finite positive linear combinations of unitaries induced by the group elements. The positivity of the coefficients is needed so that the condition involving the determinant implies the nilpotency. However, these conditions are no longer equivalent if one allows negative coeficients. As a consequence, this new definition of normative pair extends the number of languages that can be defined.

Corollary 15. *Let $\mathfrak{S}$ be the group of finite permutations over* $\mathbf{N}$*, and for all* $n \in \mathbf{N}$*,* $\mathfrak{N}_n = \mathfrak{N}$*. Then* $(\mathfrak{N}_0, \mathfrak{S})$ *is a normative pair in* $(\bigotimes_{s \in S} \mathfrak{N}_s) \rtimes_{\hat{\alpha}} \mathfrak{S}$*.*

In this particular case, the algebra $(\bigotimes_{n \in \mathbf{N}} \mathfrak{N}_n) \rtimes_{\alpha} \mathfrak{S}$ is the type II_1 hyperfinite factor. This is one of the reason why Girard considered it, as it is then possible to use Fuglede-Kadison determinant. From now on, we will consider this normative pair fixed, and we will study two sets of observations.

Definition 16 ($P_{\geqslant 0}$ and P_+)**.** An observation $(\phi_{i,j})_{0 \leqslant i,j \leqslant 6k} \in \mathfrak{M}_6(\mathfrak{S}) \otimes \mathfrak{M}_k(\mathbf{C})$ is said to be:

- *positive* when for all i, j, $\phi_{i,j}$ is a positive finite linear combination of unitaries induced by $\mathfrak{S}$, i.e. $\phi_{i,j} = \sum_{l=0}^{m} \alpha_l \lambda(g_l)$ with $\alpha_l \geqslant 0$;
- *boolean* when for all i, j, $\phi_{i,j}$ is a finite sum of unitaries induced by $\mathfrak{S}$, i.e. $\phi_{i,j} = \sum_{l=0}^{m} \alpha_l \lambda(g_l)$ with $\alpha_l = 1$.

We then define the following sets of observations:

$$\begin{array}{rcl} P_{\geqslant 0} & = & \{\phi \mid \phi \text{ is a positive observation}\} \\ P_{+} & = & \{\phi \mid \phi \text{ is a boolean observation}\} \end{array}$$

5. Non-Deterministic Pointer Machines

We define in this section the notion of non-deterministic pointer machine, an abstract device that may only move a fixed number of pointers, read the pointed values and according to its non-determinist transition function change the position of the pointers and its state. It may remind of PURPLE (Schöpp & Hofmann 2008) as we cannot remember any value nor access to the address of the pointers, and it may be interesting to study the relations between the latter and our machines. However, since this paper is focused on the study of a non-deterministic framework, we postpone this question to a future work dealing with deterministic complexity classes.

Our device will take as input a binary integer $n \in \mathbf{N}$ encoded by a string $a_0, \dots, a_s \in \{0, 1, \star\}^{s+1}$, with $a_i \neq \star$ for $1 \leqslant i \leqslant s$ and $a_0 = \star$. We will moreover consider this input to be circular, that is $a_0 = a_{s+1}$, and to be stored on a finite *input tape* where the address i of a value a_i will be denoted by $\#a_i$.

Pointers usually store an address $\#a$ and allows to access to the pointed value a. Here, the pointers we manipulate are differents: they are associated to "memory cells" where the value a is stored, but this value is updated *only when the pointer moves*. We don't have access to the address $\#a$ and a is read and stored only when a pointer arrives at $\#a$. Those restrictions are due to the simulation we will explain in the following section.

Definition 17 (Instructions)**.** Let p be a pointer. We will denote by $\#p$ the value of the pointer (the address it points at). We define the following instructions:

- $p+$ which means "move the pointer forward", i.e. let $\#p$ be $\#p + 1$;
- $p-$ which means "move the pointer backward", i.e. let $\#p$ be $\#p - 1$;
- ϵ_i which means "do not move the pointer".

When p moves the new value at $\#p$ is read and stored.

Definition 18 (Non-Deterministic Pointer Machines)**.** A non-deterministic pointer machine (NDPM) with $p \in \mathbf{N}$ pointers is a triplet $M = \{Q, \Sigma, \to\}$ where

— Q is the set of *states*, $Q = \{\mathbf{q}_0, \mathbf{q}_1, \dots, \mathbf{q}_e\}$;
— $\Sigma = \{0, 1, \star\}$ is the *alphabet*;
— $\rightarrow \subseteq (\Sigma^p \times Q) \times (\mathscr{P}((P^p \times Q) \setminus \emptyset) \cup \{\textbf{accept, reject}\})$ is the binary transition relation.

where $\mathscr{P}$ denotes the power set and P is the set of instructions $\{p_i+, p_i-, \epsilon_i \mid i \in \{1, \dots, p\}\}$.

A configuration of a NDPM M is as usual a "snapshot" of M at a given time: it is an element of $\mathbf{N}^p \times \Sigma^p \times Q$ which encodes the addresses $\#p_1, \dots, \#p_p$ of the pointers, $a_{i_1}, \dots, a_{i_p} \in \Sigma^p$ the last values read by the pointers and $\mathbf{q} \in Q$ the current state.
We define a *pseudo-configuration* c as "partial snapshot" of a NDPM M: $c \in C_M$ gives the last values read by the p pointers and the current state, *but does not give the addresses of the p pointers*. Formally, the set of pseudo-configurations is defined as $C_M = \Sigma^p \times Q$, which is the domain of the transition relation: this relation is such that for all $c \in C_M$ there exists a r such that $c \rightarrow r$ and r is either p instructions and a new configuration, either **accept** or **reject**.

Let M a NDPM, $c \in C_M$ and n an input. We define $M_c(n)$ as M with n encoded as a string on its input tape starting in the pseudo-configuration c with its p pointers at $\#a_0$. As the pointers did not moved yet, M do not know the values they point at, but c gives this information, not necessarily in a faithful way. An entry n is *accepted* (resp. *rejected*) by M with pseudo-configuration $c \in C_M$ if after a finite number of transitions every branch of $M_c(n)$ reaches **accept** (resp. at least a branch reaches **reject**). We say that $M_c(n)$ halts if it accepts or rejects n. We will say that M decides a set S if there exists a pseudo-configuration $c \in C_M$ such that $M_c(n)$ accepts if and only if $n \in S$.

Shorthands We use the symbol $*$ for any symbol in Σ, 0/1 for "0 or 1". For instance $(*, 0, \mathbf{q}) \rightarrow (\epsilon_1, p_2+, \mathbf{q}')$ will be a shorthand for

$$\begin{array}{rcl} (0,0,\mathbf{q}) & \rightarrow & (\epsilon_1, p_2+, \mathbf{q}') \\ (1,0,\mathbf{q}) & \rightarrow & (\epsilon_1, p_2+, \mathbf{q}') \\ (\star,0,\mathbf{q}) & \rightarrow & (\epsilon_1, p_2+, \mathbf{q}') \end{array}$$

One movement at a time We can prove that for all $M = \{Q, \Sigma, \rightarrow\}$ with p pointers recognizing a set there exists $M' = \{Q', \Sigma, \rightarrow'\}$ with p pointers recognizing the same set such that for all $\sigma_1, \dots, \sigma_p, \mathbf{q} \rightarrow' p_1, \dots, p_p, \mathbf{q}'$ at most one instruction among $p_1, \dots, p_p$ differs from ϵ_i. For instance we can translate $(\star, 0, 1, \mathbf{q}) \rightarrow (p_1+, p_2-, p_3+, \mathbf{q}')$ – for $\mathbf{q}_{p_2}$ and $\mathbf{q}_{p_3}$ two "fresh" states, not in Q – into

$$\begin{array}{rcl} (\star,0,1,\mathbf{q}) & \rightarrow' & (p_1+, \epsilon_2, \epsilon_3, \mathbf{q}_{p_2}) \\ (*,*,*,\mathbf{q}_{p_2}) & \rightarrow' & (\epsilon_1, p_2-, \epsilon_3, \mathbf{q}_{p_3}) \\ (*,*,*,\mathbf{q}_{p_3}) & \rightarrow' & (\epsilon_1, \epsilon_2, p_3+, \mathbf{q}') \end{array}$$

The number of states of M' and the number of transitions needed by M' to decides the same set increase, but that does not affect our machine in terms of complexity as the number of transitions and the cardinality of K will not be measures of the complexity of ours machines.

The only worry about complexity will be what sets we can recognize with this framework. It turns out that we can recognize a **Co-NL**-complete set as we will see in the following.

Definition 19 (s-t-Conn-COMP). We define the following problem: "given a (directed) graph encoded as a list of adjacences, accept if and only if there is **no** path from the source (numbered 1) to the target (numbered n) in the graph". This problem is known as **s-t-Conn-COMP** or **REACHABILITY-COMP**, is **Co-NL** complete and was shown to be in **NL** (Immerman 1988), and so is also **NL**-complete. We define the set

$$\textbf{s-t-Conn-COMP} = \{n \in \mathbf{N} \mid n \text{ does } \textbf{not} \text{ encode a graph where there is a path from 1 to } n\}$$

Proposition 20. *There exists a NDPM M that decides* ***s-t-Conn-COMP****.*

Proof. Given a graph of size n, the input will be

$$\star\underbrace{00\dots00}_{n\text{bits}}\boxed{1}\overbrace{a_{11}0a_{12}0\dots0a_{1n-1}0a_{1n}}^{\text{edges going from1}}\boxed{1}\dots\boxed{1}\overbrace{a_{n1}0a_{n2}0\dots0a_{nn-1}0a_{nn}}^{\text{edges going from } n}\boxed{1}$$

where (a_{ij}) is the adjacency matrix, that is to say that $a_{ij} = 1$ if and only if there is an edge from the node numbered by i to the node numbered by j, 0 elsewhere. The boxed bits in the figure above are "separating" bits, between the coding of n and the list of adjacences, and between the coding of the edges going from i to the coding of the edges going from $i+1$.

We define a NDPM M such that $M_c(n)$ with $c = \{\star,\star,\star,\star,\textbf{Init}\}$ accepts if and only if $n \in$ **s-t-Conn-COMP**.

The transition relation of M is presented in the figure 1. Informally, our algorithm goes as follow: p_1 counts the size of the path followed. Every time we follow an edge, we move p_1 forward on the string made of n bits (second line of 10). p_2 will scan the encoding of the outgoing edges of a node, "followed" by p_3: when p_2 is reading a_{ij} then p_3 will be at a_{j1}. If $a_{ij} = 1$ (premise of 10), a non-determinist transition takes place: on one way we continue to scan the outgoing edges from i, on the other we increment p_1, place p_2 at a_{j1} and p_3 at a_{11}. p_4 "follows" p_3 on the n first bits, and if p_4 reaches a 1 when p_2 reads that there is an edge, it means that there is an edge whose target is n, and so we reject (11). When p_2 finishes to browse the adjacency list of an edge, we accept (6). If p_1 reaches a 1 and p_4 reads a 0 (premise of 12), it means that we already followed n edges without ever targeting the node n, so we end up accepting. As we know that if there is a path from 1 to n then there exists a path of size at most n, $M_c(n)$ will accept if and only if $n \in$**s-t-Conn-Comp**, elsewhere $M_c(n)$ rejects. □

It turns out that all NDPMs cannot be represented as operators. Indeed, Lemma 25 needs an additional requirement: acyclicity. However, as we will now show, a language which is decided by a non-deterministic pointer machine is decided by an acyclic NDPM.

Definition 21 (Acyclicity)**.** A NDPM M is said to be *acyclic* when for all $c \in C_M$ and all entry $n \in \mathbf{N}$, $M_c(n)$ halts.

$$(\star,\star,\star,\star,\mathbf{Init}) \to (p_1+,p_2+,p_3+,p_4+,\mathbf{Init}) \tag{1}$$
$$(*,0,*,*,\mathbf{Init}) \to (\epsilon_1,p_2+,p_3+,\epsilon_4,\mathbf{Init}) \tag{2}$$
$$(*,1,*,*,\mathbf{Init}) \to (\epsilon_1,p_2+,\epsilon_3,\epsilon_4,\mathbf{out.edge?}) \tag{3}$$
$$(*,0,*,*,\mathbf{out.edge?}) \to (\epsilon_1,p_2+,\epsilon_3,p_4+,\mathbf{no.edge}) \tag{4}$$
$$(*,0,*,*,\mathbf{no.edge}) \to (\epsilon_1,\epsilon_2,p_3+,\epsilon_4,\mathbf{p3.next.node}) \tag{5}$$
$$(*,1,*,*,\mathbf{no.edge}) \to \mathbf{accept} \tag{6}$$
$$(*,*,*,*,\mathbf{p3.next.node}) \to (\epsilon_1,\epsilon_2,p_3+,\epsilon_4,\mathbf{reading.sep.bit}) \tag{7}$$
$$(*,*,0,*,\mathbf{reading.sep.bit}) \to (\epsilon_1,\epsilon_{,}p_3+,\epsilon_4,\mathbf{p3.next.node}) \tag{8}$$
$$(*,*,1,*,\mathbf{reading.sep.bit}) \to (\epsilon_1,p_2+,\epsilon_3,\epsilon_4,\mathbf{out.edge?}) \tag{9}$$
$$(*,1,*,*,\mathbf{out.edge?}) \to \begin{cases}(\epsilon_1,p_2+,\epsilon_3,p_4+,\mathbf{no.edge})\\(p_1+,\epsilon_2,\epsilon_3,p_4+,\mathbf{edge.found})\end{cases} \tag{10}$$
$$(*,*,*,1,\mathbf{edge.found}) \to \mathbf{reject} \tag{11}$$
$$(1,*,*,0,\mathbf{edge.found}) \to \mathbf{accept} \tag{12}$$
$$(*,*,*,0,\mathbf{edge.found}) \to (\epsilon_1,p_2-,\epsilon_3,p_4-,\mathbf{rewind.p2.p4}) \tag{13}$$
$$(*,*,*,0/1,\mathbf{rewind.p2.p4}) \to (\epsilon_1,p_2-,\epsilon_3,p_4-,\mathbf{rewind.p2.p4}) \tag{14}$$
$$(*,*,*,\star,\mathbf{rewind.p2.p4}) \to (\epsilon_1,p_2-,\epsilon_3,\epsilon_4,\mathbf{rewind.p2}) \tag{15}$$
$$(*,0/1,*,*,\mathbf{rewind.p2}) \to (\epsilon_1,p_2-,\epsilon_3,\epsilon_4,\mathbf{rewind.p2}) \tag{16}$$
$$(*,\star,*,*,\mathbf{rewind.p2}) \to (\epsilon_1,p_2+,p_3-,\epsilon_4,\mathbf{exchange.p2.p3.}) \tag{17}$$
$$(*,*,0/1,*,\mathbf{exchange.p2.p3}) \to (\epsilon_1,p_2+,p_3-,\epsilon_4,\mathbf{exchange.p2.p3.}) \tag{18}$$
$$(*,*,\star,*,\mathbf{exchange.p2.p3}) \to (\epsilon_1,\epsilon_2,p_3+,\epsilon_4,\mathbf{get.p3.to.start}) \tag{19}$$
$$(*,*,0,*,\mathbf{get.p3.to.start}) \to (\epsilon_1,\epsilon_2,p_3+,\epsilon_4,\mathbf{get.p3.to.start}) \tag{20}$$
$$(*,*,1,*,\mathbf{get.p3.to.start}) \to (\epsilon_1,p_2+,\epsilon_3,\epsilon_4,\mathbf{out.edge?}) \tag{21}$$

Figure 1. The transition relation to decide **s-t-Conn-COMP**

Lemma 22. *For all NDPM M that decides a set S there exists an acyclic NDPM M' that decides S.*

Proof. To prove this, we need to prove that for all $n \in \mathbf{N}$ and $c \in C_M$ there exists a $c' \in C_{M'}$ such that if $M_c(n)$ does not halt then $M'_{c'}(n)$ rejects, and if $M_c(n)$ accepts (resp. rejects) then $M'_{c'}(n)$ accepts (resp. rejects).

We know that the number of configurations of M is bounded by $O(n^c)$ for c a constant depending of p, the number of pointers of M. So we know that if M does more than $O(n^c)$ transitions, it will never halt. To obtain M' we will simply add $c+1$ pointers that will behave like the hands of a clock. The first one moves forward each time we make a transition. Each time the i-th one has traveled through the whole input tape, the $i+1$-th one moves forward. When the last one reaches the beginning of the input tape, M' accepts. It ensures us that M' has made $O(n^{c+1})$ transitions: if M' did not halt after this time, it means it was in a loop.

We set $p' = p + c + 1$, and for all $\mathbf{q} \in Q$, every time we had in M the transition:

$$(\vec{i}, \mathbf{q}) \to (\vec{m}, \mathbf{q}')$$

we add to $\to' \in M'$ the following set of transitions (for $p+1 \leqslant k < p'$):

$$(\vec{i}, \star, \ldots, \star, \mathbf{q}) \to' (\vec{m}, p_{p+1}+, \ldots, p_{p'}+, \mathbf{q'})$$
$$(\vec{i}, 0/1, \ldots, 0/1, \mathbf{q}) \to' (\vec{m}, p_{p+1}+, \sigma_{p+2}, \ldots, \sigma_{p'}, \mathbf{q'})$$
$$(\vec{i}, 0/1, \ldots, 0/1, i_k = \star, 0/1, \ldots, 0/1, \mathbf{q}) \to' (\vec{m}, \sigma_{p+1}, \ldots, \sigma_{k-1}, p_k+, p_{k+1}+, \sigma_{k+2}, \ldots, \sigma_{p'}, \mathbf{q'})$$
$$(\vec{i}, 0/1, \ldots, 0/1, \star, \mathbf{q}) \to' \mathbf{reject}$$

Then, for all $c' = (\vec{i}, p_{p+1}, \ldots, p_{p'}) \in C_{M'}$ that does not appear on the left-hand side in the previous set of transitions, we add $c' \to' \mathbf{reject}$.

For all $c = (m_1, \ldots, m_p, \mathbf{q}) \in C_M$ we define $c^t = (m_1, \ldots, m_p, \star, \ldots, \star, \mathbf{q}) \in C_{M'}$.

Now take a pseudo-configuration $c \in C_M$, several cases arise:

— If $M_c(n)$ was halting, it was in less than $O(n^c)$ transitions so $M'_{c^t}(n)$ will have the same behavior.
— If $M_c(n)$ was entering a loop, $M'_{c^t}(n)$ rejects after $O(n^{c+1})$ transitions.

However, since we supposed that M was deciding S, we know there exists a pseudo-configuration $c_0 \in C_M$ such that for all $n \in \mathbf{N}$, $M_{c_0}(n)$ halts, hence never enters a loop. As a result, by considering the pseudo-configuration c_0^t we can see that M' will decide the set S. Moreover it is clear that for all $c' \in C_{M'}$ and all $n \in \mathbf{N}$, $M'_{c'}(n)$ always halt, so M' is acyclic. □

Definition 23. Let {NDPM} (resp. {ANDPM}) be the class of sets S such that there exists a NDPM (resp. an acyclic NDPM) that decides S.

Proposition 24.

$$\textbf{co-NL} \subseteq \{ANDPM\}$$

Proof. Proposition 20 shows that **co-NL** ⊆ {NDPM}. Moreover, it is clear that {ANDPM} ⊆ {NDPM} and the preceding lemma shows that {NDPM} ⊆ {ANDPM}. As a consequence, we have {NDPM} = {ANDPM} and thus **co-NL** ⊆ {ANDPM}. □

6. Encoding Non-Deterministic Pointer Machines

6.1. *Encoding a Machine*

Our aim in this section is to prove (Theorem 25) that for any acyclic NDPM M and pseudo-configuration $c \in C_M$, there exists an observation $M_c^\bullet \in \mathfrak{M}_6(\mathfrak{O}) \otimes \mathfrak{Q}_M$ such that for all $N_n \in \mathfrak{M}_6(\mathfrak{N})$ a binary representation of n, $M_c(n)$ accepts if and only if $M_c^\bullet(N_n \otimes 1_{\mathfrak{Q}_M})$ is nilpotent.

We will define $M_c^\bullet$ as an operator of $\mathfrak{M}_6(\mathfrak{S}) \otimes \mathfrak{Q}_M$, where

$$\mathfrak{Q}_M = \underbrace{\mathfrak{M}_6(\mathbf{C}) \otimes \mathfrak{M}_6(\mathbf{C}) \otimes \cdots \otimes \mathfrak{M}_6(\mathbf{C})}_{p \text{ times}} \otimes \mathfrak{M}_k(\mathbf{C})$$

The intuition is that the i-th copy of $\mathfrak{M}_6(\mathbf{C})$ represents a "memory block" that contains the

last value read by the i-th pointer. We will therefore distinguish for each copy of $\mathfrak{M}_6(\mathbf{C})$ a basis $(0o, 0i, 1o, 1i, s, e)$ corresponding to the different values that a pointer can read. The last algebra in the tensor product represent a set of states: we will distinguish a basis $Q \cup B$ where Q is the set of states of the machine M and B is an additional set of states needed for the definition of $M_c^\bullet$. To sum up, the distinguished basis of $\mathfrak{Q}_M$ considered will be denoted by tuples $(a_1, \dots, a_n, \mathbf{q})$. Notice that such a tuple naturally corresponds to a pseudo-configuration when $q \in Q$.

As a consequence of the tensoring of N_n with the unit of the algebra of states, the integer is considered at the same time in every possible pseudo-configuration. As a result, the computation represented by the sequence $M_c^\bullet(N_n \otimes 1_{\mathfrak{Q}_M}), (M_c^\bullet(N_n \otimes 1_{\mathfrak{Q}_M}))^2, \dots$ somehow simulates all the computations $M_c(n)$ (c a pseudo configuration) simultaneously. However, the representation of reject cannot be done without considering an initial pseudo-configuration, something that will be explained in the next subsection.

The main difficulty is now to encode the transition relation. In order to do this, we will encode each couple $(c, t) \in \rightarrow$ by an operator $\phi_{c,t}$. The encoding of the transition relation will then correspond to the sum:

$$\rightarrow^\bullet = \sum_{c \in C_M} \sum_{t \text{ s.t. } c \rightarrow t} \phi_{c,t}$$

Before explaining the encoding of basic operations, we first define the projections π_{0o}, π_{0i}, π_{1o}, π_{1i}, π_{start}, π_{end} of $\mathfrak{M}_6(\mathbf{C})$ as the projections onto the subspace generated by the distinguished basis. We moreover define $\pi_{0\cdot} = \pi_{0i} + \pi_{0o}$ and $\pi_{1\cdot} = \pi_{1o} + \pi_{1i}$ to identify the bit currently read without considering if we come from the left (the output) or the right (the input).

For the sake of simplicity, we also define the following operators in $\mathfrak{Q}_M$: if $c = (a_1, \dots, a_p, \mathbf{q})$ and $c' = (a'_1, \dots, a'_p, \mathbf{q}')$, we define the partial isometry:

$$(c \rightarrow c') = (a_1 \rightarrow a'_1) \otimes \cdots \otimes (a_p \rightarrow a'_p) \otimes (\mathbf{q} \rightarrow \mathbf{q}')$$

where

$$(p \rightarrow p') = \begin{array}{cc} & p \\ p' & \begin{pmatrix} 0 & \dots & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \iddots & \vdots \\ 0 & \dots & 1 & \dots & 0 \\ \vdots & \iddots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & \dots & 0 \end{pmatrix} \end{array} \quad \begin{array}{l} (p \in \{a_1, \dots, a_p, \mathbf{q}\}) \\ (p' \in \{a'_1, \dots, a'_p, \mathbf{q}'\}) \end{array}$$

For $\mathbf{S}$ a set of states, we will use the notation $(\mathbf{S} \rightarrow a'_i)$ (denoted $(\rightarrow a'_i)$ when $\mathbf{S}$ contains all possible states) for the element that goes from any state in $\mathbf{S}$ to a'_i, which is defined as $\sum_{s \in \mathbf{S}} (s \rightarrow a'_i)$.

A transition that impacts only on the values stored in the subset $p_{i_1}, \dots, p_{i_l}$ and the state $\mathbf{q}$ will be denoted by

$$([a_{i_1} \rightarrow a'_{i_1}]_{i_1}; \dots; [a_{i_l} \rightarrow a'_{i_l}]_{i_l}; \mathbf{q} \rightarrow \mathbf{q'}) = u_1 \otimes u_2 \otimes \cdots \otimes u_p \otimes (\mathbf{q} \rightarrow \mathbf{q'})$$

where $u_i = (a_{i_j} \rightarrow a'_{i_j})$ if $\exists j, i = i_j$, $u_i = \text{Id}$ elsewhere, and $\mathbf{q} \rightarrow \mathbf{q}' = \text{Id}$ if $\mathbf{q} = \mathbf{q}'$.

We are now ready to define the operators needed to encode the basic operations of the machine. Considering the von Neumann algebra $\mathfrak{M}_6(\mathfrak{R}) \otimes \mathfrak{Q}_M$ as $\mathfrak{M}_6(\mathbf{C}) \otimes \mathfrak{R} \otimes \mathfrak{Q}_M$, we will define these operators as tensor products $u \otimes v \otimes w$, where $u \in \mathfrak{M}_6(\mathbf{C})$, $v \in \mathfrak{S} \subset \mathfrak{R}$ and $w \in \mathfrak{Q}_M$.

6.2. *Basic Operations*

From now on, we consider given a machine M and a pseudo-configuration $c = (a_1, \dots, a_p, \mathbf{q})$.

6.2.1. *Move forward (resp. backward) a pointer, read a value and change state.* We want to encode the action "move forward (resp. backward) the pointer k when we are in the pseudo-configuration $c = (a_1, \dots, a_p; \mathbf{q})$, read the value a'_k stored at $\#p_k$ and change the pseudo-configuration for $c' = (a_1, \dots, a_{k-1}, a'_k, a_{k+1}, \dots, a_p; \mathbf{q}')$". For $a \in \{f, b\}, j \in \{0, 1, \star\}$, and[§§] $d = ji$ (resp. $d = jo$) if $a = f$ (resp. $a = b$), we define the operators:

$$\begin{array}{rcl} m_k(c; \mathbf{q} \to \mathbf{q}') & = & 1 \otimes \tau_{0,k} \otimes ([s \in \{0,1,\star\} \to sd]; \mathbf{q} \to \mathbf{mouv}_k) \\ la_{k,j}(c; \mathbf{q} \to \mathbf{q}') & = & \pi_d \otimes \tau_{0,k} \otimes ([\to \pi_d]_k; \mathbf{mouv}_k \to \mathbf{q}') \end{array}$$

where $[s \in \{0,1,\star\} \to si]$ (resp. $[s \in \{0,1,\star\} \to so]$) is the operator $\sum_{s \in \{0,1,\star\}} (si \to si) + (so \to si)$ (resp. $\sum_{s \in \{0,1,\star\}} (si \to so) + (so \to so)$).

The operator $m_k(c; \mathbf{q} \to \mathbf{q}')$ somehow selects the chosen pointer while $la_{k,j}(c; \mathbf{q} \to \mathbf{q}')$ allows to move forward or backward according to a when the next bit is j. Then the operators that encode the forward and backward moves are respectively:

$$\begin{array}{rcl} \alpha_k(c; \mathbf{q} \to \mathbf{q}') & = & m_k(c; \mathbf{q} \to \mathbf{q}') + \sum_{i \in \{0,1,\text{end}\}} lf_{k,i}(c; \mathbf{q} \to \mathbf{q}') \\ \rho_k(c; \mathbf{q} \to \mathbf{q}') & = & m_k(c; \mathbf{q} \to \mathbf{q}') + \sum_{i \in \{0,1,\text{start}\}} lb_{k,i}(c; \mathbf{q} \to \mathbf{q}') \end{array}$$

6.2.2. *Accept.* The case of acceptance is especially easy: we want to stop the computation, so every transition $(a_1, \dots, a_n; \mathbf{q}) \to \mathbf{accept}$ will be encoded by 0.

6.2.3. *Reject.* We want the operator to loop to simulate the reject of the machine. Indeed, a rejection must ensure that the resulting operator $M_c^\bullet(N_n \otimes 1)$ will not be nilpotent. A first naive attempt:

$$\text{reject}_{\text{naive}} = \text{Id}_{\mathfrak{M}_6(\mathbf{C})} \otimes \text{Id} \otimes \pi_{reject}$$

shows that it is possible to make the computation loop, as $N_n^k \neq 0$ for all $k \in \mathbf{N}$.

$$((N_n \otimes 1_{\mathfrak{E}}) \text{Id}_{\mathfrak{M}_6(\mathbf{C})} \otimes \text{Id} \otimes \pi_{reject})^k = (N_n \otimes \pi_{reject})^k = N_n^k \otimes \pi_{reject}$$

Indeed, as $\to^\bullet$ is built as a sum of the basic operations, $\text{reject}_{\text{naive}}$ appears in it, and so $M^\bullet(N_n \otimes 1_{\mathfrak{E}})$ cannot be nilpotent[¶¶].

So we have to be a little more clever to insure the operator will loop if *and only if* the

[§§] We consider that $\star o = \text{start}$ and $\star i = \text{end}$.
[¶¶] Remember that $N_n \otimes 1_{\mathfrak{E}} = N_n \otimes \text{Id}_{\bigotimes_{n=1}^{n+1} \mathfrak{M}_6(\mathbf{C})} \otimes \pi_{reject} + N_n \otimes \text{Id}_{\bigotimes_{n=1}^{n+1} \mathfrak{M}_6(\mathbf{C})} \otimes (1 - \pi_{reject})$.

operator that simulates the reject is reached. To do that, we simply make the operator go back to the chosen pseudo-configuration c when it reaches this operator. In this way, if reject was reached after applying the machine with a pseudo-configuration c', we enforce the computation of the machine on c. As a consequence, if the integer was accepted by the machine in state c, the rejection that corresponds to a computation on c' will be temporary: once rejection attained, the computation restarts with pseudo-configuration c and will therefore halts accepting.

To encode this, we add two states to the machine — $\mathbf{back}_i$ and $\mathbf{mouv\text{-}back}_i$ — for each $i = 1,\dots,p$, and we define:

$$\begin{aligned}
rm_i &= 1\otimes\tau_{0,i}\otimes(\mathbf{back}_i\to\mathbf{mouv\text{-}back}_i)\\
rr_i &= \pi_{0o}+\pi_{1o}\otimes\tau_{0,i}\otimes([\to\pi_{0o}+\pi_{1o}]_i;\mathbf{mouv\text{-}back}_i\to\mathbf{back}_i)\\
rc_i &= \pi_{start}\otimes\tau_{0,i}\otimes([\to a_i]_i;\mathbf{mouv\text{-}back}_i\to\mathbf{back}_{i+1})\quad(1\leqslant i<p)\\
rc_p &= \pi_{start}\otimes\tau_{0,p}\otimes([\to a_p]_p;\mathbf{mouv\text{-}back}_p\to\mathbf{q}_0)
\end{aligned}$$

The operator simulating the reject by making the operator loop is then $\mathbf{r} = \sum_{i=1}^{p} rm_i + rr_i + rc_i$. The transitions of the form $(a_1,\dots,a_n;\mathbf{q})\to\mathbf{reject}$ will then be encoded by the operator that represents the "transition" $(a_1,\dots,a_n;\mathbf{q})\to(a_1,\dots,a_n;\mathbf{back}_0)$.

6.3. *First Inclusions*

Lemma 25. *Let M be an acyclic NDPM, $c\in C_M$ and $M_c^\bullet$ the encoding we just defined. For all $n\in\mathbf{N}$ and every binary representation $N_n\in\mathfrak{M}_6(\mathfrak{N})$ of n:*

$$M_c(n) \textit{ accepts} \Leftrightarrow M_c^\bullet(N_n\otimes 1) \textit{ is nilpotent}$$

Proof. Let us fix $n\in\mathbf{N}$ and N_n one of its binary representations. Considering the representation of the reject it is clear that if a branch of $M_c(n)$ rejects, the operator $M_c^\bullet(N_n\otimes 1)$ will not be nilpotent, so we just have to prove that if $M_c(n)$ accepts then $M_c^\bullet(N_n\otimes 1)$ is nilpotent. We prove its reciprocal: let's suppose $M_c^\bullet(N_n\otimes 1)$ is not nilpotent. In this product N_n is given to the operator $M_c^\bullet$ that starts the simulation of the computation of M with input n in every possible pseudo-configuration at the same time. Since the encoding of M takes in argument a pseudo-configuration $c\in C_M$, we know that there exists a j such that $M_c^\bullet(N_n\otimes 1)\pi_j$ is the simulation of $M_c(n)$, but the computation takes place in the other projections too: for $i\neq j$ it is possible that $M_c^\bullet(N_n\otimes 1)\pi_i$ loops where for a k $(M_c^\bullet(N_n\otimes 1))^k\pi_j = 0$. We can correct this behavior thanks to the acyclicity: if $M_c^\bullet(N_n\otimes 1)$ is not nilpotent it is because at some point the **reject** state has been reached. After this state of reject is reached (let's say after $r\in\mathbf{N}$ iterations) we know that $M_c^\bullet(N_n\otimes 1)^r\pi_i$ is exactly the simulation of $M_c(n)$. If it loops again, it truly means that $M_c(n)$ rejects. So we just proved that $M_c^\bullet(N_n\otimes 1)$ is not nilpotent if and only if $(M_c^\bullet(N_n\otimes 1))^k\pi_j\neq 0$ for all $k\in\mathbf{N}$. But it is clear that in this case M with pseudo-configuration c rejects the entry n. ☐

Proposition 26.

$$\textbf{\textit{co-NL}} \subseteq \{ANDPM\} \subseteq \{P_+\} \subseteq \{P_{\geqslant 0}\}$$

Proof. The first inclusion is given by Proposition 24. By Lemma 25, we have {ANDPM} ⊆

$\{P_+\}$ since the representation $M_c^\bullet$ of a couple (M,c), where M is an acyclic NDPM and $c \in C_M$, is obviously in P_+. Moreover, since $P_+ \subset P_{\geqslant 0}$, we have $\{P_+\} \subseteq \{P_{\geqslant 0}\}$. ☐

7. Positive observations and co-NL

To show that $\{P_{\geqslant 0}\}$ is included in **co-NL**, we will show that the product of a binary representation and an observation in $P_{\geqslant 0}$ is the image of a matrix by an injective morphism. This fact was used by Girard, but we felt it needed to be more precisely stated and proved in the following (quite technical) lemma.

Lemma 27. *We consider the normative pair $(\mathfrak{N}_0, \mathfrak{S})$ defined in Corollary 15 and denote by $\mathfrak{K}$ the algebra $(\bigotimes_{n\geqslant 0}\mathfrak{R}) \rtimes \mathfrak{S}$. Let N_n be a binary representation of an integer n in $\mathfrak{M}_6(\mathfrak{N}_0)$ and $\Phi \in \mathfrak{M}_6(\mathfrak{S})$ be an observation in $P_{\geqslant 0}$. Then there exists an integer k, an injective morphism $\psi : \mathfrak{M}_k(\mathbf{C}) \to \mathfrak{K}$ and two matrices $M \in \mathfrak{M}_6(\mathfrak{M}_k(\mathbf{C}))$ and $\bar{\Phi} \in \mathfrak{M}_6(\mathfrak{M}_k(\mathbf{C})) \otimes \mathfrak{E}$ such that $Id \otimes \psi(M) = (N_n \otimes 1_{\mathfrak{E}})$ and $Id \otimes \psi \otimes Id_{\mathfrak{E}}(\bar{\Phi}) = \Phi$.*

Proof. We denote by n the integer represented by N_n and $R \in \mathfrak{M}_{6(n+1)}(\mathbf{C})$ its matricial representation. Then there exists a morphism $\theta : \mathfrak{M}_{n+1}(\mathbf{C}) \to \mathfrak{R}$ such that $\mathrm{Id} \otimes \theta(R) = N_n$ by Proposition 8. Composing θ with the inclusion $\mu : \mathfrak{M}_{n+1}(\mathbf{C}) \to \bigotimes_{n=0}^{N}\mathfrak{M}_{n+1}(\mathbf{C})$, $x \mapsto x \otimes 1 \otimes \cdots \otimes 1$, we get:

$$\mathrm{Id} \otimes (\bigotimes_{n=0}^{N} \theta(\mu(R)) = \bar{N}_n \otimes \underbrace{1 \otimes \cdots \otimes 1}_{N \text{ copies}}$$

where $\bar{N}_n$ is the representation of n in $\mathfrak{M}_6(\mathbf{C}) \otimes \mathfrak{R}$ (recall the representation N_n in the statement of the lemma is an element of $\mathfrak{M}_6(\mathbf{C}) \otimes \mathfrak{K}$).

Moreover, since Φ is an observation, it is contained in the subalgebra induced by the subgroup $\mathfrak{S}_N$ where N is a fixed integer, i.e. the subalgebra of $\mathscr{S}$ generated by $\{\lambda(\sigma) \mid \sigma \in \mathfrak{S}_N\}$. We thus consider the algebra $(\bigotimes_{n=0}^{N}\mathfrak{M}_{n+1}(\mathbf{C})) \rtimes \mathfrak{S}_N$. It is isomorphic to a matrix algebra $\mathfrak{M}_k(\mathbf{C})$: the algebra $\bigotimes_{n=0}^{N}\mathfrak{M}_{n+1}(\mathbf{C})$ can be represented as an algebra of operators acting on the Hilbert space $\mathbf{C}^{N(n+1)}$, and the crossed product $(\bigotimes_{n=0}^{N}\mathfrak{M}_{n+1}(\mathbf{C})) \rtimes \mathfrak{S}_N$ is then defined as a subaglebra $\mathfrak{I}$ of the algebra $\mathscr{L}(L^2(\mathfrak{S}_N, \mathbf{C}^{(n+1)^N})) \cong \mathfrak{M}_{(n+1)^N N!}(\mathbf{C})$. We want to show that $(N_n \otimes 1_{\mathfrak{E}})$ and Φ are the images of matrices in $\mathfrak{I}$ by an injective morphism ψ which we still need to define.

Let us denote by α the action of $\mathfrak{S}_N$ on $\bigotimes_{n=0}^{N}\mathfrak{M}_{n+1}(\mathbf{C})$. By definition, $\mathfrak{I} = (\bigotimes_{n=0}^{N}\mathfrak{M}_{n+1}(\mathbf{C})) \rtimes \mathfrak{S}_N$ is generated by two families of unitaries:

— $\lambda_\alpha(\sigma)$ where $\sigma \in \mathfrak{S}_N$;
— $\pi_\alpha(x)$ where x is an element of $\bigotimes_{n=0}^{N}\mathfrak{M}_{n+1}(\mathbf{C})$.

We will denote by γ the action of $\mathfrak{S}$ on $\bigotimes_{n=0}^{\infty}\mathfrak{R}$. Then $\mathfrak{K} = (\bigotimes_{n\geqslant 0}\mathfrak{R}) \rtimes \mathfrak{S}$ is generated by the following families of unitaries:

— $\lambda_\gamma(\sigma)$ for $\sigma \in \mathfrak{S}$;
— $\pi_\gamma(x)$ for $x \in \bigotimes_{n\geqslant 0}\mathfrak{R}$.

As we already recalled, Φ is an observation in $P_{\geqslant 0}$ and is thus contained in the subalgebra induced by the subgroup $\mathfrak{S}_N$. Moreover, N_n is the image through θ of an element of $\mathfrak{M}_{n+1}(\mathbf{C})$. Denoting β the action of $\mathfrak{S}_N$ on $\bigotimes_{n=0}^{N}\mathfrak{R}$, the two operators we are interested in are elements of the subalgebra $\mathfrak{J}$ of $\mathfrak{K}$ generated by:

— $\lambda_\beta(\sigma)$ for $\sigma \in \mathfrak{S}_N$;

— $\pi_\beta(\bigotimes_{n=0}^{N}\theta(x))$ for $x \in \bigotimes_{n=0}^{N}\mathfrak{M}_{n+1}(\mathbf{C})$.

We recall that Φ is a matrix whose coefficients are finite positive linear combinations of elements $\lambda_\gamma(\sigma)$ where $\sigma \in \mathfrak{S}_N$, i.e. (denoting by k the dimension of the algebra of states):

$$\Phi = (\sum_{i\in I_{a,b}} \alpha^i_{a,b}\lambda_\gamma(\sigma^i_{a,b}))_{1\leqslant a,b\leqslant 6k}$$

We can therefore associate to Φ the matrix $\bar{\Phi}$ defined as $\bar{\Phi} = (\sum_{i\in I_{a,b}} \alpha^i_{a,b}\lambda_\alpha(\sigma^i_{a,b}))_{1\leqslant a,b\leqslant 6k}$. We will now use the theorem stating the crossed product algebra does not depend on the chosen representation (Theorem 5). The algebra $\bigotimes_{n=0}^{N}$ is represented (faithfully) by the morphism $\pi_\beta\circ\bigotimes_{n=0}^{\infty}\theta$. We deduce from this that there exists an isomorphism from $\mathfrak{I}$ to the algebra generated by the unitaries $\lambda_\beta(\sigma)$ ($\sigma \in \mathfrak{S}_N$) and $\pi_\beta\circ\bigotimes_{n=0}^{\infty}\theta(x)$ ($x \in \bigotimes_{n=0}^{N}\mathfrak{M}_{n+1}(\mathbf{C})$). This isomorphism induces an injective morphism ω from $\mathfrak{I}$ into $\mathfrak{J}$ such that:

$$\begin{array}{rcl}\omega(\pi_\alpha(x)) & = & \pi_\beta(\bigotimes_{n=0}^{N}\theta(x)) \\ \omega(\lambda_\alpha(\sigma)) & = & \lambda_\beta(\sigma)\end{array}$$

We will denote by ι the inclusion map $\bigotimes_{n=0}^{N}\mathfrak{R} \subset \bigotimes_{n=0}^{\infty}\mathfrak{R}$ and υ the inclusion map $\mathfrak{S}_N \subset \mathfrak{S}$. We will once again use the same theorem as before, but its application is not as immediate as it was. Let us denote by $\mathfrak{S}_N\backslash\mathfrak{S}$ the set of the orbits of $\mathfrak{S}$ for the action of $\mathfrak{S}_N$ by multiplication on the left, and let us chose a representant $\bar{\tau}$ in each of these orbits. Recall the set of orbits is a partition of $\mathfrak{S}$ and that $\mathfrak{S}_N \times \mathfrak{S}_N\backslash\mathfrak{S}$ is in bijection with $\mathfrak{S}$. As a consequence, the Hilbert space $L^2(\mathfrak{S}_N, L^2(\mathfrak{S}_N\backslash\mathfrak{S}, \bigotimes_{n=0}^{\infty}\mathbb{H}))$ is unitarily equivalent to $L^2(\mathfrak{S}, \bigotimes_{n=0}^{\infty}\mathbb{H})$. We will therefore represent $\bigotimes_{n=0}^{N}\mathfrak{R}$ on this Hilbert space and show this representation corresponds to π_γ. For each $x \in \bigotimes_{n=0}^{N}\mathfrak{R}$, we define $\rho(x)$ by:

$$\rho(x)\xi(\bar{\tau}) = \gamma(\bar{\tau}^{-1})(\iota(x))\xi(\bar{\tau})$$

This representation is obviously faithful. We can then define the crossed product of this representation with the group $\mathfrak{S}_N$ on $L^2(\mathfrak{S}_N, L^2(\mathfrak{S}_N\backslash\mathfrak{S}, \bigotimes_{n=0}^{\infty}\mathbb{H}))$. The resulting algebra is generated by the operators (in the following, ξ is an element of the Hilbert space $L^2(\mathfrak{S}_N, L^2(\mathfrak{S}_N\backslash\mathfrak{S}, \bigotimes_{n=0}^{\infty}\mathbb{H}))$):

$$\begin{array}{rcl}\lambda(\nu)\xi(\bar{\tau})(\sigma) & = & \xi(\bar{\tau})(\nu^{-1}\sigma) \\ \pi(x)\xi(\bar{\tau})(\sigma) & = & \rho(\beta(\sigma^{-1})(x))\xi(\bar{\tau})(\sigma) \\ & = & \gamma(\bar{\tau}^{-1})(\gamma(\sigma^{-1})(\iota(x)))\xi(\bar{\tau})(\sigma) \\ & = & \gamma(\bar{\tau}^{-1}\sigma^{-1})(\iota(x)))\xi(\bar{\tau})(\sigma) \\ & = & \gamma((\sigma\bar{\tau})^{-1})(\iota(x)))\xi(\bar{\tau})(\sigma)\end{array}$$

Through the identification of $L^2(\mathfrak{S}_N, L^2(\mathfrak{S}_N\backslash\mathfrak{S}, \bigotimes_{n=0}^{\infty}\mathbb{H}))$ and $L^2(\mathfrak{S}, \bigotimes_{n=0}^{\infty}\mathbb{H}))$, we therefore

$$\begin{tikzcd}
\mathfrak{S}_N \arrow[r, "\lambda_\alpha"] \arrow[d, equal] & (\bigotimes_{n=0}^{N} \mathfrak{M}_{n+1}(\mathbf{C})) \rtimes_\alpha \mathfrak{S}_N \arrow[d, "\omega"'] & \bigotimes_{n=0}^{N} \mathfrak{M}_{n+1}(\mathbf{C}) \arrow[l, "\pi_\alpha"'] \arrow[d, "\bigotimes_{n=0}^{N}\theta"] \\
\mathfrak{S}_N \arrow[r, "\lambda_\beta"] \arrow[d, "\subset"'] & (\bigotimes_{n=0}^{N} \mathfrak{R}) \rtimes_\beta \mathfrak{S}_N \arrow[d, "\zeta"'] & \bigotimes_{n=0}^{N} \mathfrak{R} \arrow[l, "\pi_\beta"'] \arrow[d, "\iota"] \\
\mathfrak{S} \arrow[r, "\lambda_\gamma"'] & (\bigotimes_{n\geqslant 0} \mathfrak{R}) \rtimes_\gamma \mathfrak{S} & \bigotimes_{n=0}^{\infty} \mathfrak{R} \arrow[l, "\pi_\gamma"]
\end{tikzcd}$$

Figure 2. Representation of the main morphisms defined in the proof of Lemma 27

get (where $\xi \in L^2(\mathfrak{S}_N, L^2(\mathfrak{S}_N\backslash\mathfrak{S}, \bigotimes_{n=0}^{\infty}\mathbb{H})))$:

$$\begin{aligned}
\lambda(\nu)\xi(\sigma\bar{\tau}) &= \xi(\nu^{-1}\sigma\bar{\tau}) \\
&= \lambda_\gamma(\nu)\xi(\sigma\bar{\tau}) \\
\pi(x)\xi(\sigma t\bar{a}u) &= \gamma((\sigma\bar{\tau})^{-1})(\iota(x)))\xi(\sigma\bar{\tau}) \\
&= \pi_\gamma(\iota(x))\xi(\sigma\bar{\tau})
\end{aligned}$$

Applying theorem 5 we finally get the existence of an injective morphism ζ from $\mathfrak{J}$ into $\mathfrak{K}$ such that:

$$\begin{aligned}
\pi_\beta(x) &\mapsto \pi_\gamma(\iota(x)) \\
\lambda_\beta(\sigma) &\mapsto \lambda_\gamma(\sigma)
\end{aligned}$$

Figure 2 illustrates the situation. We now define $\psi : \mathfrak{J} \to \mathfrak{K}$ by $\psi = \zeta \circ \omega$. Noticing that $N_n = \mathrm{Id}_{\mathfrak{M}_6(\mathbf{C})} \otimes (\pi_\gamma(\iota \circ \mu(\bar{N}_n))$, we get:

$$\begin{aligned}
\mathrm{Id}_{\mathfrak{M}_6(\mathbf{C})} \otimes \psi(M) &= \mathrm{Id}_{\mathfrak{M}_6(\mathbf{C})} \otimes \psi(\mathrm{Id} \otimes \pi_\alpha(\mathrm{Id} \otimes \mu)(R)) \\
&= \mathrm{Id}_{\mathfrak{M}_6(\mathbf{C})} \otimes \pi_\gamma(\iota \circ \bigotimes_{n=0}^{N} \theta(\mu(R))) \\
&= \mathrm{Id}_{\mathfrak{M}_6(\mathbf{C})} \otimes \pi_\gamma(\iota(\bar{N}_n \otimes 1 \otimes \cdots \otimes 1)) \\
&= \mathrm{Id}_{\mathfrak{M}_6(\mathbf{C})} \otimes \pi_\gamma(\iota \circ \mu(\bar{N}_n)) \\
&= N_n
\end{aligned}$$

$$\begin{aligned}
\mathrm{Id}_{\mathfrak{M}_6(\mathbf{C})} \otimes \psi \otimes \mathrm{Id}_{\mathfrak{E}}(\bar{\Phi}) &= (\sum_{i \in I_{a,b}} \alpha^i_{a,b} \psi(\lambda_\alpha(\sigma^i_{a,b})))_{1 \leqslant a,b \leqslant 6k} \\
&= (\sum_{i \in I_{a,b}} \alpha^i_{a,b} \lambda_\gamma(\sigma^i_{a,b}))_{1 \leqslant a,b \leqslant 6k} \\
&= \Phi
\end{aligned}$$

The (injective) morphism ψ thus satisfies all the required properties. $\square$

We are now read to prove the last inclusion to get the main theorem.

Proposition 28. $\{P_{\geqslant 0}\} \subseteq$ ***co-NL***

Proof. Let $\Phi \in P_{\geqslant 0}$, $\mathfrak{E}$ its algebra of states and N_n the representation of an integer n. By lemma 27, we know there exists a morphism ϕ (and ψ as in the lemma, $\phi = \mathrm{Id}_{\mathfrak{M}_6(\mathbf{C})} \otimes \psi \otimes \mathrm{Id}_{\mathfrak{E}}$) and two matrices M and $\bar{\Phi}$ such that $\phi(M \otimes 1_{\mathfrak{E}}) = N_n \otimes 1_{\mathfrak{E}}$ and $\phi(\bar{\Phi}) = \Phi$. So we have $\Phi(N_n \otimes 1_{\mathfrak{E}})$ nilpotent if and only if $(M \otimes 1_{\mathfrak{E}})\bar{\Phi}$ nilpotent. Our aim is now to prove that checking the nilpotency of this matrix is in **co-NL**.
Our algebra is:

$$\mathfrak{M}_6(\mathbf{C}) \otimes ((\underbrace{\mathfrak{M}_{n+1}(\mathbf{C}) \otimes \cdots \otimes \mathfrak{M}_{n+1}(\mathbf{C}))}_{N \text{ copies}} \rtimes \mathfrak{S}_N) \otimes \mathfrak{E}$$

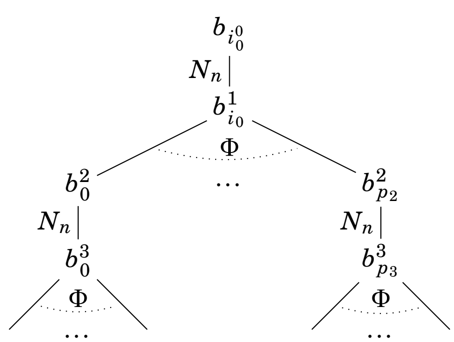
and we know an element of its basis will be of the form

$$(\pi, a_0, a_1, \ldots, a_N; \sigma; e)$$

where $\pi \in \{1,6\}$ is an element of the basis $(0o, 0i, 1o, 1i, s, e)$ of $\mathfrak{M}_6(\mathbf{C})$, $a_i \in \{1,\ldots,N\}$ are the elements of the basis chosen to represent the integer n, $\sigma \in \mathfrak{S}_N$ and e is an element of a basis of $\mathfrak{E}$. When we apply $M \otimes 1_{\mathfrak{E}}$ representing the integer to an element of this basis, we obtain one and only one vector of the basis $(\pi, a_0, a_1, \ldots, a_N; \sigma; e)$. When we apply to this element the observation $\bar{\Phi}$ we obtain a linear positive combination of $L \in \mathbf{N}$ elements of the basis:

$$\bar{\Phi}(\pi, a_0, a_1, \ldots, a_N; \sigma; e) = \sum_{i=0}^{L} \alpha_i (\rho, a_{\tau_i(0)}, \ldots, a_{\tau_i(N)}; \tau_i \sigma; e_i)$$

With a non-deterministic machine, we can follow the computation in parallel on each basis vector thus obtained. The computation can then be regarder as a tree (denoting by b_i^j the elements of the basis encountered):



We know that L and the nilpotency degree of $\bar{\Phi}(M \otimes 1_{\mathfrak{E}})$ are both bounded by the dimensions of the underlying space, that is to say $6(n+1)^N N!k$ where k is the dimension of $\mathfrak{E}$. Since every coefficient α_i is positive, the matrix is thus nilpotent if and only if every branch of this tree is of length at most $6(n+1)^N N!k$.

We only have a logarithmic amount of information to store (the current basis vector), and every time a branch splits a non-deterministic transition takes place in order to continue the computation on every branch. $\square$

Theorem 29.

$$\{ANDPM\} = \{P_+\} = \{P_{\geqslant 0}\} = \textbf{\textit{co-NL}}$$

Proof. By combining Proposition 26 and Proposition 28. □

8. Conclusion and Perspectives

This work explains the motivations and choices made by Girard when he proposed this new approach to the study of complexity classes. In particular, we explained how the representation of integers by matrices is an abstraction of sequent calculus proofs of the type of binary lists in **ELL**, and how the use of the hyperfinite factor allows to overcome the lack of uniformity of the matrix representation. We then introduced a notion of normative pair which differs from the one introduced by Girard and showed how the crossed product construction can be used to define such pairs. The change from an interaction based on the determinant to one relying on nilpotency allows to consider a larger class of groups in the construction based on the crossed product. Moreover, even if the two definitions are equivalent in some cases, such as the one considered in this paper, they differ in some others.

We then introduced non-deterministic pointer machines as a technical tool to show that **co-NL** $\subseteq \{P_+\}$. The proof of this inclusion, which was only sketched in Girard's paper, helps get more insights on how computation is represented by operators. Moreover, it gives a new characterization of **co-NL** in term of machines. We then proved that $\{P_{\geqslant 0}\} \subseteq$ **co-NL** following the proof given by Girard (Girard 2011*b*), providing a proper statement and a proof of the key technical result that was not provided by Girard.

We believe that this new approach of complexity can be used to characterize other complexity classes. Two different possibilities should be considered: changing the normative pair, and changing the set of observations. As we showed, one could define a normative pair from a group action by using the crossed product construction. However, obtaining new results in this way requires to overcome the difficulty of finding appropriate groups.

The second possibility, which seems at the time less complicated would be to consider other sets of observations for the same normative pair. For instance, one could define the set of observations whose coefficients are unitaries induced by group elements and whose norm is equal to 1 (so that there are at most one non-zero coefficient in each column). Denoting this set by P_1, we can easily adapt the proof of Proposition 28 to show that $\{P_1\} \subseteq$ **L**. However, the question of wether the corresponding class $\{P_1\}$ is equal or strictly included in **L**, and its eventual relations to PURPLE (Schöpp & Hofmann 2008), still need to be answered.

References

Aubert, C. (2011), Sublogarithmic uniform boolean proof nets, *in* J.-Y. Marion, ed., 'DICE', Vol. 75 of *EPTCS*, pp. 15–27.

Baillot, P., Pedicini, M. et al. (2001), 'Elementary complexity and geometry of interaction', *Fundamenta Informaticae* **45**(1-2), 1–31.

Conway, J. B. (1990), *A course in functional analysis*, Springer.

Dal Lago, U. & Hofmann, M. (2009), 'Bounded linear logic, revisited', *Typed Lambda Calculi and Applications* pp. 80–94.

Danos, V. & Joinet, J.-B. (2001), 'Linear logic & elementary time', *Information and Computation* **183**, 2003.

Girard, J.-Y. (1989), Towards a geometry of interaction, *in* 'Proceedings of the AMS Conference on Categories, Logic and Computer Science'.

Girard, J.-Y. (2011*a*), 'Geometry of interaction V: Logic in the hyperfinite factor.', *Theoretical Computer Science* **412**, 1860–1883.

Girard, J.-Y. (2011*b*), 'Normativity in logic', *Epistemology vs. Ontology*.

Haagerup, U. (1975), 'The standard form of von neumann algebras', *Math. Scand.*

Immerman, N. (1988), Nondeterministic space is closed under complementation, *in* 'Structure in Complexity Theory Conference, 1988. Proceedings., Third Annual', IEEE, pp. 112–115.

Lafont, Y. (2004), 'Soft linear logic and polynomial time', *Theoretical Computer Science* **318**(1), 163–180.

Lago, U. (2005), The geometry of linear higher-order recursion, *in* 'Logic in Computer Science, 2005. LICS 2005. Proceedings. 20th Annual IEEE Symposium on', IEEE, pp. 366–375.

Murphy, G. J. (1990), *C^*-algebras and operator theory*, Academic Press Inc., Boston, MA.

Schöpp, U. (2007), Stratified bounded affine logic for logarithmic space, *in* 'Logic in Computer Science, 2007. LICS 2007. 22nd Annual IEEE Symposium on', IEEE, pp. 411–420.

Schöpp, U. & Hofmann, M. (2008), Pointer programs and undirected reachability, *in* 'Electronic Colloquium on Computational Complexity (ECCC)', Vol. 15, p. 090.

Seiller, T. (2011), 'Interaction graphs: Multiplicatives'.

Seiller, T. (2012*a*), 'Interaction graphs: Additives', *Arxiv preprint arXiv:1205.6557*.

Seiller, T. (2012*b*), Logique dans le facteur hyperfini : géometrie de l'interaction et complexité, PhD thesis, Université de la Méditerranée.

Takesaki, M. (2001), *Theory of Operator Algebras 1*, Vol. 124 of *Encyclopedia of Mathematical Sciences*, Springer.

Takesaki, M. (2003*a*), *Theory of Operator Algebras 2*, Vol. 125 of *Encyclopedia of Mathematical Sciences*, Springer.

Takesaki, M. (2003*b*), *Theory of Operator Algebras 3*, Vol. 127 of *Encyclopedia of Mathematical Sciences*, Springer.

Terui, K. (2004), Proof Nets and Boolean Circuits, *in* 'Proceedings of LICS'04', pp. 182–191.